SPEECH PROCESSING SYSTEM

The present invention relates to an apparatus for and method of audio encoding. The invention particularly relates to a statistical processing of an input speech signal to derive parameter values defining the speech production system which generated the speech and the subsequent transmission of those parameter values.

A number of speech transmission systems have been proposed. These can be categorised into systems which transmit essentially the speech signals and those which parameterise the speech first (based on some speech model) and transmit the parameters. The problem with the first technique is that it requires a relatively large bandwidth to transmit the speech signals compared with parameterising the speech signals first and then transmitting the parameters. However, parameterising the speech results in more complex transmitter circuitry in order to determine the parameter values and more complex receiver circuitry in order to regenerate the speech from the transmitted parameters. Further, the second technique also reduces the quality of the regenerated speech at the receiver, since some speech information

09565535.053001

will inevitably be lost through the parameterisation.

Many different techniques are known to parameterise speech signals. One of the most commonly used techniques is based on a linear prediction analysis of the speech. With this technique, the entire speech signal is divided into a number of time frames (typically having a duration of 10 to 30 ms) and a set of linear prediction parameters (or coefficients) is calculated to represent the speech within each time frame. This linear prediction analysis assumes that the value of a current speech sample can be predicted from a linear weighted combination of the k most recent speech samples. Based on this model, the task of the linear prediction analyses is to identify the value of the weightings (or coefficients) which minimises the mean squared error between the actual value of the current speech sample and the predicted value of the current speech sample.

One of the problems with this linear prediction analysis is that is analyses the speech within each frame in isolation from the speech within other frames. It also assumes that the same number of weightings or coefficients represent the speech within each time frame. As a result, errors are introduced into the

09865585.053001

representation.

An aim of the present invention is to provide an alternative technique for parameterising speech prior to transmission from a transmitting terminal to a receiving terminal.

According to one aspect, the present invention provides an audio encoding system comprising: a memory for storing a probability density function for parameters of a predetermined audio model which is assumed to have generated a set of received audio signal values; means for applying the set of received audio signal values to the probability density function; means for processing said function with the set of received audio signal values applied to determine samples of parameter values from said probability density function; means for analysing at least some of the determined samples to determine parameter values that are representative of the received audio signal values; and means for encoding said determined parameter values to generate encoded data representative of the received audio signal values.

Exemplary embodiments of the present invention will now be described with reference to the accompanying drawings

in which:

Figure 1 is a schematic block diagram illustrating the principal components of a speech encoding and transmission system;

Figure 2a is a flow chart illustrating the processing steps performed by the transmission side of the system shown in Figure 1;

Figure 2b is a flow chart illustrating the processing steps performed by the receiving side of the system shown in Figure 1;

Figure 3 is a block diagram representing a model employed by a statistical analysis unit which forms part of the speech transmission system shown in Figure 1;

Figure 4 is a flow chart illustrating the processing steps performed by a model order selection unit forming part of the statistical analysis unit shown in Figure 1;

Figure 5 is a flow chart illustrating the main processing steps employed by a Simulation Smoother which forms part of the statistical analysis unit shown in Figure 1;

Figure 6 is a block diagram illustrating the main processing components of the statistical analysis unit shown in Figure 1;

Figure 7 is a memory map illustrating the data that is stored in a memory which forms part of the statistical analysis unit shown in Figure 1;

Figure 8 is a flow chart illustrating the main processing steps performed by the statistical analysis unit shown in Figure 6;

Figure 9a is a histogram for a model order of an auto regressive filter model which forms part of the model shown in Figure 3;

Figure 9b is a histogram for the variance of process noise modelled by the model shown in Figure 3; and

Figure 9c is a histogram for a third coefficient of the AR filter model.

Embodiments of the present invention can be implemented in computer hardware, computer software or a mix of computer hardware and software. When implemented using

computer software, the program instructions that make the programmable hardware operate in accordance with the present invention may be supplied on, for example, a storage device such as a magnetic disc or by downloading the software from a computer device over a computer network.

Figure 1 shows a speech encoding and transmission system. Electrical signals representative of input speech from the microphone 7 are input to a filter 15 which removes unwanted frequencies (in this embodiment frequencies above 8 kHz). The filtered signal is then sampled (at a rate of 16 kHz) and digitized by an analogue to digital converter 17 and the digitized speech samples are then stored in a buffer 19. Sequential blocks (or frames) of speech samples are then passed from the buffer 19 to a statistical analyses unit 21 which performs a statistical analysis of each frame of speech samples in sequence to determine, amongst other things, a set of auto regressive (AR) coefficients representative of the speech within the frame. In this embodiment, the AR coefficients output by the statistical analysis unit 21 are input to a channel encoder which encodes the sequences of AR filter coefficients so that they are in a more suitable form for transmission through a communications channel. The

encoded AR filter coefficients are then passed to a transmitter 73 where the encoded data is used to modulate a carrier signal which is then transmitted to a remote receiver 75. The receiver 75 demodulates the received signal to recover the encoded data which is then decoded by a decoder 76. The sequence of AR coefficients output by the decoder are then either passed to a speech recognition unit 77 which compares the sequences with stored reference models (not shown) to generate a recognition result or to a speech synthesis unit 79 which regenerates the speech and outputs it via a loudspeaker 81. As shown, prior to application to the speech synthesis unit 79, the sequences of AR coefficients may also pass through an optional processing unit 83 (shown in phantom) which can be used to manipulate the characteristics of the speech that is synthesised.

Figure 2a is a flow chart illustrating the processing steps performed by the channel encoder 71 of the system shown in Figure 1. As shown, in step S101 the channel encoder 71 receives the speech parameters and the quality indicator for the current segment of speech to be transmitted. The processing then proceeds to step S103 where the channel encoder 71 determines whether or not the speech parameters to be transmitted were generated

from a high quality speech signal. If they are, then the processing proceeds to step S105 where the channel encoder 71 encodes the speech parameters using an efficient encoding technique which minimises the data to be transmitted. If, on the other hand, step S103 determines that the speech parameters were derived from low quality speech, then the processing proceeds to step S107 where the channel encoder 71 uses a less efficient encoding technique (or no encoding) which minimises information lost in the encoding. After step S105 or S107, the processing proceeds to step S109 where the channel encoder 71 outputs the data to be transmitted to the transmitter unit 73 which transmits the encoded speech to the receiver 75. In this embodiment, the channel encoder 71 also encodes and transmits the quality indicator to the receiver 75 so that the receiver can decode the encoded speech parameters using the appropriate decoding technique. In order that the receiver can always decode the encoded speech quality indicator, the channel encoder 71 encodes this information using a standard encoding technique regardless of what the quality indicator is.

Figure 2b is a flow chart illustrating the processing steps performed by the decoder 76 shown in Figure 1. As

shown, in step S111, the decoder 76 recovers the quality indicator from the received encoded speech data. The processing then proceeds to step S113 where the decoder 76 determines whether or not the received encoded speech parameters where generated from a high quality speech signal. If they were, then the processing proceeds to step S115 where the decoder uses a decoding technique corresponding to the efficient encoding technique used in step S105. If, on the other hand, the decoder 76 determines that the received encoded speech data was generated from a low quality speech signal, then the processing proceeds to step S117 where the decoder 76 decodes the received data using a decoding technique corresponding to the less efficient encoding technique used in step S107. The processing then proceeds to step S119 where the decoded speech parameters are output either to the speech recognition unit 77 or to the speech synthesis unit 79.

Statistical Analysis Unit - Theory and Overview

As mentioned above, the statistical analysis unit 21 analyses the speech within successive frames of the input speech signal. In most speech processing systems, the frames are overlapping. However, in this embodiment, the frames of speech are non-overlapping and have a duration

of 20ms which, with the 16kHz sampling rate of the analogue to digital converter 17, results in a frame size of 320 samples.

In order to perform the statistical analysis on each of the frames, the analysis unit 21 assumes that there is an underlying process which generated each sample within the frame. The model of this process used in this embodiment is shown in Figure 2. As shown, the process is modelled by a speech source 31 which generates, at time t = n, a raw speech sample s(n). Since there are physical constraints on the movement of the speech articulators, there is some correlation between neighbouring speech samples. Therefore, in this embodiment, the speech source 31 is modelled by an auto regressive (AR) process. In other words, the statistical analysis unit 21 assumes that a current raw speech sample (s(n)) can be determined from a linear weighted combination of the most recent previous raw speech samples, i.e.:

$$s(n) = a_1 s(n-1) + a_2 s(n-2) + \ldots\ldots + a_k s(n-k) + e(n) \qquad (1)$$

where a_1, a_2......a_k are the AR filter coefficients representing the amount of correlation between the speech samples; k is the AR filter model order; and e(n)

represents random process noise which is involved in the generation of the raw speech samples. As those skilled in the art of speech processing will appreciate, these AR filter coefficients are the same coefficients that the linear prediction analysis estimates albeit using a different processing technique.

As shown in Figure 3, the raw speech samples s(n) generated by the speech source are input to a channel 33 which models the acoustic environment between the speech source 31 and the output of the analogue to digital converter 17. Ideally, the channel 33 should simply attenuate the speech as it travels from the source 31 to the microphone. However, due to reverberation and other distortive effects, the signal (y(n)) output by the analogue to digital converter 17 will depend not only on the current raw speech sample (s(n)) but it will also depend upon previous raw speech samples. Therefore, in this embodiment, the statistical analysis unit 21 models the channel 33 by a moving average (MA) filter, i.e.:

$$y(n) = h_0 s(n) + h_1 s(n-1) + h_2 s(n-2) + \dots\dots + h_r s(n-r) + \varepsilon(n) \qquad (2)$$

where y(n) represents the signal sample output by the analogue to digital converter 17 at time t = n; h_0, h_1,

$h_2 \ldots h_r$ are the channel filter coefficients representing the amount of distortion within the channel 33; r is the channel filter model order; and $\varepsilon(n)$ represents a random additive measurement noise component.

For the current frame of speech being processed, the filter coefficients for both the speech source and the channel are assumed to be constant but unknown. Therefore, considering all N samples (where N = 320) in the current frame being processed gives:

$$\begin{aligned}
s(n) &= a_1 s(n-1) + a_2 s(n-2) + \ldots\ldots + a_k s(n-k) + e(n) \\
s(n-1) &= a_1 s(n-2) + a_2 s(n-3) + \ldots\ldots + a_k s(n-k-1) + e(n-1) \\
&\vdots \\
s(n-N+1) &= a_1 s(n-N) + a_2 s(n-N-1) + \ldots\ldots + a_k s(n-k-N+1) + e(n-N+1)
\end{aligned} \qquad (3)$$

which can be written in vector form as:

$$\underline{s}(n) = S.\underline{a} + \underline{e}(n) \qquad (4)$$

where

$$S = \begin{bmatrix} s(n-1) & s(n-2) & s(n-3) & \cdots & s(n-k) \\ s(n-2) & s(n-3) & s(n-4) & \cdots & s(n-k-1) \\ s(n-3) & s(n-4) & s(n-5) & \cdots & s(n-k-2) \\ \vdots & & & \ddots & \\ s(n-N) & s(n-N-1) & s(n-N-2) & \cdots & s(n-k-N+1) \end{bmatrix}_{Nxk}$$

and

$$\underline{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ . \\ . \\ . \\ a_k \end{bmatrix}_{kx1} \qquad \underline{s}(n) = \begin{bmatrix} s(n) \\ s(n-1) \\ s(n-2) \\ . \\ . \\ . \\ s(n-N+1) \end{bmatrix}_{Nx1} \qquad \underline{e}(n) = \begin{bmatrix} e(n) \\ e(n-1) \\ e(n-2) \\ . \\ . \\ . \\ e(n-N+1) \end{bmatrix}_{Nx1}$$

As will be apparent from the following discussion, it is also convenient to rewrite equation (3) in terms of the random error component (often referred to as the residual) e(n). This gives:

$$\begin{aligned} e(n) &= s(n) - a_1 s(n-1) - a_2 s(n-2) - \ldots\ldots - a_k s(n-k) \\ e(n-1) &= s(n-1) - a_1 s(n-2) - a_2 s(n-3) - \ldots\ldots - a_k s(n-k-1) \\ &\vdots \\ e(n-N+1) &= s(n-N+1) - a_1 s(n-N) - a_2 s(n-N-1) - \ldots\ldots - a_k s(n-k-N+1) \end{aligned} \tag{5}$$

which can be written in vector notation as:

$$\underline{e}(n) = \ddot{A}\,\underline{s}(n) \tag{6}$$

where

$$\ddot{A} = \begin{bmatrix} 1 & -a_1 & -a_2 & -a_3 & . & . & . & -a_k & 0 & 0 & 0 & . & . & . & 0 \\ 0 & 1 & -a_1 & -a_2 & . & . & . & -a_{k-1} & -a_k & 0 & 0 & . & . & . & 0 \\ 0 & 0 & 1 & -a_1 & . & . & . & -a_{k-2} & -a_{k-1} & -a_k & 0 & . & . & . & 0 \\ . & & & & & \ddots & & & & & & & & & \\ . & & & & & & & \ddots & & & & & & & \\ . & & & & & & & & & \ddots & & & & & \\ 0 & & & & & & & & & & & & & & 1 \end{bmatrix}_{NxN}$$

Similarly, considering the channel model defined by equation (2), with $h_0 = 1$ (since this provides a more stable solution), gives:

$$\begin{aligned} q(n) &= h_1 s(n-1) + h_2 s(n-2) + .. + h_r s(n-r) + \varepsilon(n) \\ q(n-1) &= h_1 s(n-2) + h_2 s(n-3) + .. + h_r s(n-r-1) + \varepsilon(n-1) \\ &\vdots \\ q(n-N+1) &= h_1 s(n-N) + h_2 s(n-N-1) + .. + h_r s(n-r-N+1) + \varepsilon(n-N+1) \end{aligned} \tag{7}$$

(where $q(n) = y(n) - s(n)$) which can be written in vector form as:

$$\underline{q}(n) = Y.\underline{h} + \underline{\varepsilon}(n) \tag{8}$$

where

$$Y = \begin{bmatrix} s(n-1) & s(n-2) & s(n-3) & \cdots & s(n-r) \\ s(n-2) & s(n-3) & s(n-4) & \cdots & s(n-r-1) \\ s(n-3) & s(n-4) & s(n-5) & \cdots & s(n-r-2) \\ \vdots & & & \ddots & \vdots \\ s(n-N) & s(n-N-1) & s(n-N-2) & \cdots & s(n-r-N+1) \end{bmatrix}_{Nxr}$$

and

$$\underline{h} = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ \vdots \\ h_r \end{bmatrix}_{rx1} \qquad \underline{q}(n) = \begin{bmatrix} q(n) \\ q(n-1) \\ q(n-2) \\ \vdots \\ q(n-N+1) \end{bmatrix}_{Nx1} \qquad \underline{\varepsilon}(n) = \begin{bmatrix} \varepsilon(n) \\ \varepsilon(n-1) \\ \varepsilon(n-2) \\ \vdots \\ \varepsilon(n-N+1) \end{bmatrix}_{Nx1}$$

In this embodiment, the analysis unit 21 aims to determine, amongst other things, values for the AR filter coefficients ($\underline{a}$) which best represent the observed signal samples ($\underline{y}$(n)) in the current frame. It does this by determining the AR filter coefficients ($\underline{a}$) that maximise the joint probability density function of the speech model, channel model, speech samples and the noise statistics given the observed signal samples output from the analogue to digital converter 17, i.e. by determining:

$$\max_{\underline{a}} \left\{ p(\underline{a}, k, \underline{h}, r, \sigma_e^2, \sigma_\varepsilon^2, \underline{s}(n) | \underline{y}(n)) \right\} \quad (9)$$

where σ_e^2 and σ_ε^2 represent the process and measurement noise statistics respectively. As those skilled in the art will appreciate, this function defines the probability that a particular speech model, channel model, raw speech samples and noise statistics generated the observed frame of speech samples ($\underline{y}$(n)) from the analogue to digital converter. To do this, the statistical analysis unit 21 must determine what this function looks like. This problem can be simplified by rearranging this probability density function using Bayes law to give:

$$\frac{p(\underline{y}(n)|\underline{s}(n),\underline{h},r,\sigma_\varepsilon^2)\,p(\underline{s}(n)|\underline{a},k,\sigma_e^2)\,p(\underline{a}|k)\,p(\underline{h}|r)\,p(\sigma_\varepsilon^2)\,p(\sigma_e^2)\,p(k)\,p(r)}{p(\underline{y}(n))} \tag{10}$$

As those skilled in the art will appreciate, the denominator of equation (10) can be ignored since the probability of the signals from the analogue to digital converter is constant for all choices of model. Therefore, the AR filter coefficients that maximise the function defined by equation (9) will also maximise the numerator of equation (10).

Each of the terms on the numerator of equation (10) will now be considered in turn.

p($\underline{s}$(n)|$\underline{a}$, k, σ_e^2)

This term represents the joint probability density function for generating the vector of raw speech samples ($\underline{s}$(n)) during a frame, given the AR filter coefficients ($\underline{a}$), the AR filter model order (k) and the process noise statistics (σ_e^2). From equation (6) above, this joint probability density function for the raw speech samples can be determined from the joint probability density function for the process noise. In particular p($\underline{s}$(n)|$\underline{a}$, k, σ_e^2) is given by:

$$p(\underline{s}(n)|\underline{a},k,\sigma_e^2) = p(\underline{e}(n)) \left|\frac{\delta \underline{e}(n)}{\delta \underline{s}(n)}\right|_{\underline{e}(n) = \underline{s}(n) - S\underline{a}} \qquad (11)$$

where p($\underline{e}$(n)) is the joint probability density function for the process noise during a frame of the input speech and the second term on the right-hand side is known as the Jacobean of the transformation. In this case, the Jacobean is unity because of the triangular form of the matrix Ä (see equations (6) above).

In this embodiment, the statistical analysis unit 21 assumes that the process noise associated with the speech source 31 is Gaussian having zero mean and some unknown variance σ_e^2. The statistical analysis unit 21 also assumes that the process noise at one time point is independent of the process noise at another time point. Therefore, the joint probability density function for the process noise during a frame of the input speech (which defines the probability of any given vector of process noise $\underline{e}$(n) occurring) is given by:

$$p(\underline{e}(n)) = (2\pi\sigma_e^2)^{-\frac{N}{2}} \exp\left[\frac{-\underline{e}(n)^T\underline{e}(n)}{2\sigma_e^2}\right] \qquad (12)$$

Therefore, the joint probability density function for a

vector of raw speech samples given the AR filter coefficients ($\underline{a}$), the AR filter model order (k) and the process noise variance (σ_e^2) is given by:

$$p(\underline{s}(n)|\underline{a},k,\sigma_e^2) = (2\pi\sigma_e^2)^{-\frac{N}{2}} \exp\left[\frac{-1}{2\sigma_e^2}\left(\underline{s}(n)^T\underline{s}(n) - 2\underline{a}^T S\underline{s}(n) + \underline{a}^T S^T S\underline{a}\right)\right] \quad (13)$$

p($\underline{y}$(n)|$\underline{s}$(n), $\underline{h}$, r, σ_ε^2)

This term represents the joint probability density function for generating the vector of speech samples ($\underline{y}$(n)) output from the analogue to digital converter 17, given the vector of raw speech samples ($\underline{s}$(n)), the channel filter coefficients ($\underline{h}$), the channel filter model order (r) and the measurement noise statistics (σ_ε^2). From equation (8), this joint probability density function can be determined from the joint probability density function for the process noise. In particular, p($\underline{y}$(n)|$\underline{s}$(n), $\underline{h}$, r, σ_ε^2) is given by:

$$p(\underline{y}(n)|\underline{s}(n),\underline{h},r,\sigma_\varepsilon^2) = p(\underline{\varepsilon}(n)) \left|\frac{\delta\underline{\varepsilon}(n)}{\delta\underline{y}(n)}\right| \underline{\varepsilon}(n) = \underline{q}(n) - Y\underline{h} \quad (14)$$

where p($\underline{\varepsilon}$(n)) is the joint probability density function for the measurement noise during a frame of the input speech and the second term on the right hand side is the Jacobean of the transformation which again has a value of

one.

In this embodiment, the statistical analysis unit 21 assumes that the measurement noise is Gaussian having zero mean and some unknown variance σ_ε^2. It also assumes that the measurement noise at one time point is independent of the measurement noise at another time point. Therefore, the joint probability density function for the measurement noise in a frame of the input speech will have the same form as the process noise defined in equation (12). Therefore, the joint probability density function for a vector of speech samples ($\underline{y}(n)$) output from the analogue to digital converter 17, given the channel filter coefficients ($\underline{h}$), the channel filter model order (r), the measurement noise statistics (σ_ε^2) and the raw speech samples ($\underline{s}(n)$) will have the following form:

$$p(\underline{y}(n)|\underline{s}(n),\underline{h},r,\sigma_\varepsilon^2) = (2\pi\sigma_\varepsilon^2)^{-\frac{N}{2}} \exp\left[\frac{-1}{2\sigma_\varepsilon^2}\left(\underline{q}(n)^T\underline{q}(n) - 2\underline{h}^T Y\underline{q}(n) + \underline{h}^T Y^T Y \underline{h}\right)\right] \quad (15)$$

As those skilled in the art will appreciate, although this joint probability density function for the vector of speech samples ($\underline{y}(n)$) is in terms of the variable $\underline{q}(n)$, this does not matter since $\underline{q}(n)$ is a function of $\underline{y}(n)$ and $\underline{s}(n)$, and $\underline{s}(n)$ is a given variable (ie known) for this probability density function.

p(a|k)

This term defines the *prior* probability density function for the AR filter coefficients ($\underline{a}$) and it allows the statistical analysis unit 21 to introduce knowledge about what values it expects these coefficients will take. In this embodiment, the statistical analysis unit 21 models this prior probability density function by a Gaussian having an unknown variance (σ_a^2) and mean vector ($\underline{\mu}_a$), i.e.:

$$p(\underline{a}|k,\sigma_a^2,\underline{\mu}_a) = (2\pi\sigma_a^2)^{-\frac{N}{2}} \exp\left[\frac{-(\underline{a}-\underline{\mu}_a)^T(\underline{a}-\underline{\mu}_a)}{2\sigma_a^2}\right] \tag{16}$$

By introducing the new variables σ_a^2 and $\underline{\mu}_a$, the prior density functions ($p(\sigma_a^2)$ and $p(\underline{\mu}_a)$) for these variables must be added to the numerator of equation (10) above. Initially, for the first frame of speech being processed the mean vector ($\underline{\mu}_a$) can be set to zero and for the second and subsequent frames of speech being processed, it can be set to the mean vector obtained during the processing of the previous frame. In this case, $p(\underline{\mu}_a)$ is just a Dirac delta function located at the current value of $\underline{\mu}_a$ and can therefore be ignored.

With regard to the prior probability density function for

the variance of the AR filter coefficients, the statistical analysis unit 21 could set this equal to some constant to imply that all variances are equally probable. However, this term can be used to introduce knowledge about what the variance of the AR filter coefficients is expected to be. In this embodiment, since variances are always positive, the statistical analysis unit 21 models this variance *prior* probability density function by an Inverse Gamma function having parameters α_a and β_a, i.e.:

$$p(\sigma_a^2|\alpha_a,\beta_a) = \frac{(\sigma_a^2)^{-(\alpha_a+1)}}{\beta_a\Gamma(\alpha_a)}\exp\left[\frac{-1}{\sigma_a^2\beta_a}\right] \qquad (17)$$

At the beginning of the speech being processed, the statistical analysis unit 21 will not have much knowledge about the variance of the AR filter coefficients. Therefore, initially, the statistical analysis unit 21 sets the variance σ_a^2 and the α and β parameters of the Inverse Gamma function to ensure that this probability density function is fairly flat and therefore non-informative. However, after the first frame of speech has been processed, these parameters can be set more accurately during the processing of the next frame of speech by using the parameter values calculated during the processing of the previous frame of speech.

p(h|r)

This term represents the *prior* probability density function for the channel model coefficients ($\underline{h}$) and it allows the statistical analysis unit 21 to introduce knowledge about what values it expects these coefficients to take. As with the prior probability density function for the AR filter coefficients, in this embodiment, this probability density function is modelled by a Gaussian having an unknown variance (σ_h^2) and mean vector (μ_h), i.e.:

$$p(\underline{h}|r,\sigma_h^2,\mu_h) = (2\pi\sigma_h^2)^{-\frac{N}{2}} \exp\left[\frac{-(\underline{h}-\mu_h)^T(\underline{h}-\mu_h)}{2\sigma_h^2}\right] \quad (18)$$

Again, by introducing these new variables, the prior density functions ($p(\sigma_h)$ and $p(\mu_h)$) must be added to the numerator of equation (10). Again, the mean vector can initially be set to zero and after the first frame of speech has been processed and for all subsequent frames of speech being processed, the mean vector can be set to equal the mean vector obtained during the processing of the previous frame. Therefore, $p(\mu_h)$ is also just a Dirac delta function located at the current value of μ_h and can be ignored.

With regard to the *prior* probability density function for the variance of the channel filter coefficients, again, in this embodiment, this is modelled by an Inverse Gamma function having parameters α_h and β_h. Again, the variance (σ_h^2) and the α and β parameters of the Inverse Gamma function can be chosen initially so that these densities are non-informative so that they will have little effect on the subsequent processing of the initial frame.

$p(\sigma_e^2)$ and $p(\sigma_\varepsilon^2)$

These terms are the *prior* probability density functions for the process and measurement noise variances and again, these allow the statistical analysis unit 21 to introduce knowledge about what values it expects these noise variances will take. As with the other variances, in this embodiment, the statistical analysis unit 21 models these by an Inverse Gamma function having parameters α_e, β_e and α_ε, β_ε respectively. Again, these variances and these Gamma function parameters can be set initially so that they are non-informative and will not appreciably affect the subsequent calculations for the initial frame.

p(k) and p(r)

These terms are the *prior* probability density functions

for the AR filter model order (k) and the channel model order (r) respectively. In this embodiment, these are modelled by a uniform distribution up to some maximum order. In this way, there is no prior bias on the number of coefficients in the models except that they can not exceed these predefined maximums. In this embodiment, the maximum AR filter model order (k) is thirty and the maximum channel model order (r) is one hundred and fifty.

Therefore, inserting the relevant equations into the numerator of equation (10) gives the following joint probability density function which is proportional to $p(\underline{a},k,\underline{h},r,\sigma_a^2,\sigma_h^2,\sigma_e^2,\sigma_\varepsilon^2,\underline{s}(n)\,|\,\underline{y}(n))$:

$$
\begin{aligned}
&(2\pi\sigma_\varepsilon^2)^{-\frac{N}{2}} \exp\left[\frac{-1}{2\sigma_\varepsilon^2}\left(\underline{q}(n)^T\underline{q}(n) - 2\underline{h}^T Y\underline{q}(n) + \underline{h}^T Y^T Y\,\underline{h}\right)\right] \\
&x\ (2\pi\sigma_e^2)^{-\frac{N}{2}} \exp\left[\frac{-1}{2\sigma_e^2}\left(\underline{s}(n)^T\underline{s}(n) - 2\underline{a}^T S\underline{s}(n) + \underline{a}^T S^T S\,\underline{a}\right)\right] \\
&x\ (2\pi\sigma_a^2)^{-\frac{N}{2}} \exp\left[\frac{-(\underline{a}-\mu_a)^T(\underline{a}-\mu_a)}{2\sigma_a^2}\right]\ x\ (2\pi\sigma_h^2)^{-\frac{N}{2}} \exp\left[\frac{-(\underline{h}-\mu_h)^T(\underline{h}-\mu_h)}{2\sigma_h^2}\right] \\
&x\ \frac{(\sigma_a^2)^{-(\alpha_a+1)}}{\beta_a\Gamma(\alpha_a)} \exp\left[\frac{-1}{\sigma_a^2\beta_a}\right]\ x\ \frac{(\sigma_h^2)^{-(\alpha_h+1)}}{\beta_h\Gamma(\alpha_h)} \exp\left[\frac{-1}{\sigma_h^2\beta_h}\right] \\
&x\ \frac{(\sigma_e^2)^{-(\alpha_e+1)}}{\beta_e\Gamma(\alpha_e)} \exp\left[\frac{-1}{\sigma_e^2\beta_e}\right]\ x\ \frac{(\sigma_\varepsilon^2)^{-(\alpha_\varepsilon+1)}}{\beta_\varepsilon\Gamma(\alpha_\varepsilon)} \exp\left[\frac{-1}{\sigma_\varepsilon^2\beta_\varepsilon}\right]
\end{aligned}
\tag{19}
$$

Gibbs Sampler

In order to determine the form of this joint probability density function, the statistical analysis unit 21 "draws samples" from it. In this embodiment, since the joint probability density function to be sampled is a complex multivariate function, a Gibbs sampler is used which breaks down the problem into one of drawing samples from probability density functions of smaller dimensionality. In particular, the Gibbs sampler proceeds by drawing random variates from conditional densities as follows:

first iteration

$$p(\underline{a},k|h^0,r^0,\sigma_e^{2^0},\sigma_\varepsilon^{2^0},\sigma_a^{2^0},\sigma_h^{2^0},\underline{s}(n)^0,\underline{y}(n)) \rightarrow \underline{a}^1,k^1$$

$$p(\underline{h},r|\underline{a}^1,k^1,\sigma_e^{2^0},\sigma_\varepsilon^{2^0},\sigma_a^{2^0},\sigma_h^{2^0},\underline{s}(n)^0,\underline{y}(n)) \rightarrow \underline{h}^1,k^1$$

$$p(\sigma_e^2|\underline{a}^1,k^1,\underline{h}^1,r^1,\sigma_\varepsilon^{2^0},\sigma_a^{2^0},\sigma_h^{2^0},\underline{s}(n)^0,\underline{y}(n)) \rightarrow \sigma_e^{2^1}$$

$$\vdots$$

$$p(\sigma_h^{2^1}|\underline{a}^1,k^1,\underline{h}^1,r^1,\sigma_\varepsilon^{2^1},\sigma_a^{2^1},\sigma_h^{2^1},\underline{s}(n)^0,\underline{y}(n)) \rightarrow \sigma_h^{2^1}$$

second iteration

$$p(\underline{a},k|\underline{h}^1,r^1,\sigma_e^{2^1},\sigma_\varepsilon^{2^1},\sigma_h^{2^1},\underline{s}(n)^1,\underline{y}(n)) \rightarrow \underline{a}^2,k^2$$

$$p(\underline{h},r|\underline{a}^2,k^2,\sigma_e^{2^1},\sigma_\varepsilon^{2^1},\sigma_a^{2^1},\sigma_h^{2^1},s(n)^1,y(n)) \rightarrow \underline{h}^2,r^2$$

$$\vdots$$

etc.

09865585.053001

where $(h^0, r^0, (\sigma_e^2)^0, (\sigma_\varepsilon^2)^0, (\sigma_a^2)^0, (\sigma_h^2)^0, \underline{s}(n)^0)$ are initial values which may be obtained from the results of the statistical analysis of the previous frame of speech, or where there are no previous frames, can be set to appropriate values that will be known to those skilled in the art of speech processing.

As those skilled in the art will appreciate, these conditional densities are obtained by inserting the current values for the given (or known) variables into the terms of the density function of equation (19). For the conditional density $p(\underline{a},k|\ldots)$ this results in:

$$p(\underline{a},k|\ldots) \propto \exp\left[\frac{-1}{2\sigma_e^2}\left(\underline{s}(n)^T\underline{s}(n) - 2\underline{a}^T S\underline{s}(n) + \underline{a}^T S^T S\,\underline{a}\right)\right] \tag{20}$$
$$x \quad \exp\left[\frac{-(\underline{a}-\underline{\mu}_a)^T(\underline{a}-\underline{\mu}_a)}{2\sigma_a^2}\right]$$

which can be simplified to give:

$$p(\underline{a},k|\ldots) \propto \exp\left[\frac{-1}{2}\left(\frac{\underline{s}(n)^T\underline{s}(n)}{\sigma_e^2} + \frac{\underline{\mu}_a^T\underline{\mu}_a}{\sigma_a^2} - 2\underline{a}^T\left[\frac{S\underline{s}(n)}{\sigma_e^2} + \frac{\underline{\mu}_a}{\sigma_a^2}\right] + \underline{a}^T\left[\frac{S^TS}{\sigma_e^2} + \frac{I}{\sigma_a^2}\right]\underline{a}\right)\right] \tag{21}$$

which is in the form of a standard Gaussian distribution having the following covariance matrix:

$$\Sigma_{\underline{a}} = \left[\frac{S^T S}{\sigma_e^2} + \frac{I}{\sigma_a^2}\right]^{-1} \quad (22)$$

The mean value of this Gaussian distribution can be determined by differentiating the exponent of equation (21) with respect to $\underline{a}$ and determining the value of $\underline{a}$ which makes the differential of the exponent equal to zero. This yields a mean value of:

$$\hat{\mu}_a = \left[\frac{S^T S}{\sigma_e^2} + \frac{I}{\sigma_a^2}\right]^{-1} \left[\frac{S^T \underline{s}(n)}{\sigma_e^2} + \frac{\mu_a}{\sigma_a^2}\right] \quad (23)$$

A sample can then be drawn from this standard Gaussian distribution to give $\underline{a}^g$ (where g is the g^{th} iteration of the Gibbs sampler) with the model order (k^g) being determined by a model order selection routine which will be described later. The drawing of a sample from this Gaussian distribution may be done by using a random number generator which generates a vector of random values which are uniformly distributed and then using a transformation of random variables using the covariance matrix and the mean value given in equations (22) and (23) to generate the sample. In this embodiment, however, a random number generator is used which generates random numbers from a Gaussian distribution having zero mean and a variance of one. This simplifies

the transformation process to one of a simple scaling using the covariance matrix given in equation (22) and shifting using the mean value given in equation (23). Since the techniques for drawing samples from Gaussian distributions are well known in the art of statistical analysis, a further description of them will not be given here. A more detailed description and explanation can be found in the book entitled "Numerical Recipes in C", by W. Press et al, Cambridge University Press, 1992 and in particular at chapter 7.

As those skilled in the art will appreciate, however, before a sample can be drawn from this Gaussian distribution, estimates of the raw speech samples must be available so that the matrix S and the vector $\underline{s}(n)$ are known. The way in which these estimates of the raw speech samples are obtained in this embodiment will be described later.

A similar analysis for the conditional density $p(\underline{h}, r|\ldots)$ reveals that it also is a standard Gaussian distribution but having a covariance matrix and mean value given by:

$$\Sigma_{\underline{h}} = \left[\frac{Y^T Y}{\sigma_\varepsilon^2} + \frac{I}{\sigma_h^2}\right]^{-1} \qquad \hat{\underline{\mu}}_h = \left[\frac{Y^T Y}{\sigma_\varepsilon^2} + \frac{I}{\sigma_h^2}\right]^{-1} \left[\frac{Y^T \underline{q}(n)}{\sigma_\varepsilon^2} + \frac{\underline{\mu}_h}{\sigma_h^2}\right] \tag{24}$$

from which a sample for $\underline{h}^g$ can be drawn in the manner described above, with the channel model order (r^g) being determined using the model order selection routine which will be described later.

A similar analysis for the conditional density $p(\sigma_e^2|\ldots)$ shows that:

$$p(\sigma_e^2|\ldots) \propto (\sigma_e^2)^{-\frac{N}{2}} \exp\left[\frac{-E}{2\sigma_e^2}\right] \quad \frac{(\sigma_e^2)^{-(\alpha_e+1)}}{\beta_e \Gamma(\alpha_e)} \exp\left[\frac{-1}{\sigma_e^2 \beta_e}\right] \qquad (25)$$

where:

$$E = \underline{s}(n)^T \underline{s}(n) - 2\underline{a}^T S \underline{s}(n) + \underline{a}^T S^T S \, \underline{a}$$

which can be simplified to give:

$$p(\sigma_e^2|\ldots) \propto (\sigma_e^2)^{-\left[\left(\frac{N}{2} + \alpha_e\right) + 1\right]} \exp\left[\frac{-1}{\sigma_e^2}\left(\frac{E}{2} + \frac{1}{\beta_e}\right)\right] \qquad (26)$$

which is also an Inverse Gamma distribution having the following parameters:

$$\hat{\alpha}_e = \frac{N}{2} + \alpha_e \quad and \quad \hat{\beta}_e = \frac{2\beta_e}{2 + \beta_e E} \qquad (27)$$

A sample is then drawn from this Inverse Gamma distribution by firstly generating a random number from a uniform distribution and then performing a transformation of random variables using the alpha and beta parameters given in equation (27), to give $(\sigma_e^2)^g$.

A similar analysis for the conditional density $p(\sigma_\varepsilon^2|\ldots)$ reveals that it also is an Inverse Gamma distribution having the following parameters:

$$\hat{\alpha}_\varepsilon = \frac{N}{2} + \alpha_\varepsilon \quad and \quad \hat{\beta}_\varepsilon = \frac{2\beta_\varepsilon}{2 + \beta_\varepsilon . E^*} \qquad (28)$$

where:

$$E^* = \underline{q}(n)^T \underline{q}(n) - 2\underline{h}^T Y \underline{q}(n) + \underline{h}^T Y^T Y \underline{h}$$

A sample is then drawn from this Inverse Gamma distribution in the manner described above to give $(\sigma_\varepsilon^2)^g$.

A similar analysis for conditional density $p(\sigma_a^2|\ldots)$ reveals that it too is an Inverse Gamma distribution having the following parameters:

$$\hat{\alpha}_a = \frac{N}{2} + \alpha_a \quad and \quad \hat{\beta}_a = \frac{2\beta_a}{2 + \beta_a . (\underline{a} - \underline{\mu}_a)^T (\underline{a} - \underline{\mu}_a)} \qquad (29)$$

A sample is then drawn from this Inverse Gamma

distribution in the manner described above to give $(\sigma_a^2)^g$.

Similarly, the conditional density $p(\sigma_h^2|\ldots)$ is also an Inverse Gamma distribution but having the following parameters:

$$\hat{\alpha}_h = \frac{N}{2} + \alpha_h \quad and \quad \hat{\beta}_h = \frac{2\beta_h}{2 + \beta_h \cdot (\underline{h} - \underline{\mu}_h)^T (\underline{h} - \underline{\mu}_h)} \tag{30}$$

A sample is then drawn from this Inverse Gamma distribution in the manner described above to give $(\sigma_h^2)^g$.

As those skilled in the art will appreciate, the Gibbs sampler requires an initial transient period to converge to equilibrium (known as burn-in). Eventually, after L iterations, the sample $(\underline{a}^L, k^L, \underline{h}^L, r^L, (\sigma_e^2)^L, (\sigma_\varepsilon^2)^L, (\sigma_a^2)^L, (\sigma_h^2)^L, s(n)^L)$ is considered to be a sample from the joint probability density function defined in equation (19). In this embodiment, the Gibbs sampler performs approximately one hundred and fifty (150) iterations on each frame of input speech and discards the samples from the first fifty iterations and uses the rest to give a picture (a set of histograms) of what the joint probability density function defined in equation (19) looks like. From these histograms, the set of AR coefficients ($\underline{a}$) which best represents the observed

speech samples ($\underline{y}(n)$) from the analogue to digital converter 17 are determined. The histograms are also used to determine appropriate values for the variances and channel model coefficients ($\underline{h}$) which can be used as the initial values for the Gibbs sampler when it processes the next frame of speech.

Model Order Selection

As mentioned above, during the Gibbs iterations, the model order (k) of the AR filter and the model order (r) of the channel filter are updated using a model order selection routine. In this embodiment, this is performed using a technique derived from "Reversible jump Markov chain Monte Carlo computation", which is described in the paper entitled "Reversible jump Markov chain Monte Carlo Computation and Bayesian model determination" by Peter Green, Biometrika, vol 82, pp 711 to 732, 1995.

Figure 4 is a flow chart which illustrates the processing steps performed during this model order selection routine for the AR filter model order (k). As shown, in step s1, a new model order (k_2) is proposed. In this embodiment, the new model order will normally be proposed as $k_2 = k_1 \pm 1$, but occasionally it will be proposed as $k_2 = k_1 \pm 2$

and very occasionally as $k_2 = k_1 \pm 3$ etc. To achieve this, a sample is drawn from a discretised Laplacian density function centred on the current model order (k_1) and with the variance of this Laplacian density function being chosen *a priori* in accordance with the degree of sampling of the model order space that is required.

The processing then proceeds to step s3 where a model order variable (MO) is set equal to:

$$MO = \max\left\{\frac{p(\underline{a}_{<1:k_2>}, k_2 \mid \ldots)}{p(\underline{a}_{<1:k_1>}, k_1 \mid \ldots)},\ 1\right\} \qquad (31)$$

where the ratio term is the ratio of the conditional probability given in equation (21) evaluated for the current AR filter coefficients ($\underline{a}$) drawn by the Gibbs sampler for the current model order (k_1) and for the proposed new model order (k_2). If $k_2 > k_1$, then the matrix S must first be resized and then a new sample must be drawn from the Gaussian distribution having the mean vector and covariance matrix defined by equations (22) and (23) (determined for the resized matrix S), to provide the AR filter coefficients ($\underline{a}_{<1:k2>}$) for the new model order (k_2). If $k_2 < k_1$ then all that is required is to delete the last ($k_1 - k_2$) samples of the $\underline{a}$ vector. If the ratio in equation (31) is greater than one, then this

implies that the proposed model order (k_2) is better than the current model order whereas if it is less than one then this implies that the current model order is better than the proposed model order. However, since occasionally this will not be the case, rather than deciding whether or not to accept the proposed model order by comparing the model order variable (MO) with a fixed threshold of one, in this embodiment, the model order variable (MO) is compared, in step s5, with a random number which lies between zero and one. If the model order variable (MO) is greater than this random number, then the processing proceeds to step s7 where the model order is set to the proposed model order (k_2) and a count associated with the value of k_2 is incremented. If, on the other hand, the model order variable (MO) is smaller than the random number, then the processing proceeds to step s9 where the current model order is maintained and a count associated with the value of the current model order (k_1) is incremented. The processing then ends.

This model order selection routine is carried out for both the model order of the AR filter model and for the model order of the channel filter model. This routine may be carried out at each Gibbs iteration. However,

this is not essential. Therefore, in this embodiment, this model order updating routine is only carried out every third Gibbs iteration.

Simulation Smoother

As mentioned above, in order to be able to draw samples using the Gibbs sampler, estimates of the raw speech samples are required to generate $\underline{s}(n)$, S and Y which are used in the Gibbs calculations. These could be obtained from the conditional probability density function $p(\underline{s}(n)|\ldots)$. However, this is not done in this embodiment because of the high dimensionality of $\underline{S}(n)$. Therefore, in this embodiment, a different technique is used to provide the necessary estimates of the raw speech samples. In particular, in this embodiment, a "Simulation Smoother" is used to provide these estimates. This Simulation Smoother was proposed by Piet de Jong in the paper entitled "The Simulation Smoother for Time Series Models", Biometrika (1995), vol 82,2, pages 339 to 350. As those skilled in the art will appreciate, the Simulation Smoother is run before the Gibbs Sampler. It is also run again during the Gibbs iterations in order to update the estimates of the raw speech samples. In this embodiment, the Simulation Smoother is run every fourth Gibbs iteration.

In order to run the Simulation Smoother, the model equations defined above in equations (4) and (6) must be written in "state space" format as follows:

$$\begin{aligned} \underline{\hat{s}}(n) &= \tilde{A}.\underline{\hat{s}}(n-1) + \underline{\hat{e}}(n) \\ y(n) &= \underline{h}^T.\underline{\hat{s}}(n-1) + \varepsilon(n) \end{aligned} \qquad (32)$$

where

$$\tilde{A} = \begin{bmatrix} a_1 & a_2 & a_3 & . & . & . & a_k & 0 & . & . & . & 0 \\ 1 & 0 & 0 & . & . & . & 0 & 0 & . & . & . & 0 \\ 0 & 1 & 0 & . & . & . & 0 & 0 & . & . & . & 0 \\ . & & & & . & & & & & & & \\ . & & & & & & . & & & & & \\ . & & & & & & & & . & & & \\ 0 & & & & & & & & & & 1 & 0 \end{bmatrix}_{rxr}$$

and

$$\underline{\hat{s}}(n) = \begin{bmatrix} \hat{s}(n) \\ \hat{s}(n-1) \\ \hat{s}(n-2) \\ . \\ . \\ . \\ \hat{s}(n-r+1) \end{bmatrix}_{rx1} \qquad \underline{\hat{e}}(n) = \begin{bmatrix} \hat{e}(n) \\ 0 \\ 0 \\ . \\ . \\ . \\ 0 \end{bmatrix}_{rx1}$$

With this state space representation, the dimensionality of the raw speech vectors ($\underline{\hat{s}}$(n)) and the process noise vectors ($\underline{\hat{e}}$(n)) do not need to be Nx1 but only have to be as large as the greater of the model orders - k and r. Typically, the channel model order (r) will be larger than the AR filter model order (k). Hence, the vector of

raw speech samples ($\underline{\hat{s}}(n)$) and the vector of process noise ($\underline{\hat{e}}(n)$) only need to be rx1 and hence the dimensionality of the matrix $\tilde{A}$ only needs to be rxr.

The Simulation Smoother involves two stages - a first stage in which a Kalman filter is run on the speech samples in the current frame and then a second stage in which a "smoothing" filter is run on the speech samples in the current frame using data obtained from the Kalman filter stage. Figure 5 is a flow chart illustrating the processing steps performed by the Simulation Smoother. As shown, in step s21, the system initialises a time variable t to equal one. During the Kalman filter stage, this time variable is run from t = 1 to N in order to process the N speech samples in the current frame being processed in time sequential order. After step s21, the processing then proceeds to step s23, where the following Kalman filter equations are computed for the current speech sample (y(t)) being processed:

$$
\begin{aligned}
w(t) &= y(t) - \underline{h}^T \underline{\hat{s}}(t) \\
d(t) &= \underline{h}^T P(t) \underline{h} + \sigma_\varepsilon^2 \\
\underline{k}_f(t) &= (\tilde{A} P(t) \underline{h}) . d(t)^{-1} \\
\underline{\hat{s}}(t+1) &= \tilde{A} \underline{\hat{s}}(t) + \underline{k}_f(t) . w(t) \\
L(t) &= \tilde{A} - \underline{k}_f(t) . \underline{h}^T \\
P(t+1) &= \tilde{A} P(t) L(t)^T + \sigma_e^2 . I
\end{aligned}
\qquad (33)
$$

where the initial vector of raw speech samples ($\hat{\underline{s}}(1)$) includes raw speech samples obtained from the processing of the previous frame (or if there are no previous frames then s(i) is set equal to zero for i < 1); P(1) is the variance of $\hat{\underline{s}}(1)$ (which can be obtained from the previous frame or initially can be set to σ_e^2); $\underline{h}$ is the current set of channel model coefficients which can be obtained from the processing of the previous frame (or if there are no previous frames then the elements of h can be set to their expected values - zero); y(t) is the current speech sample of the current frame being processed and I is the identity matrix. The processing then proceeds to step s25 where the scalar values w(t) and d(t) are stored together with the rxr matrix L(t) (or alternatively the Kalman filter gain vector $k_f(t)$ could be stored from which L(t) can be generated). The processing then proceeds to step s27 where the system determines whether or not all the speech samples in the current frame have been processed. If they have not, then the processing proceeds to step s29 where the time variable t is incremented by one so that the next sample in the current frame will be processed in the same way. Once all N samples in the current frame have been processed in this way and the corresponding values stored, the first stage of the Simulation Smoother is complete.

The processing then proceeds to step s31 where the second stage of the Simulation Smoother is started in which the smoothing filter processes the speech samples in the current frame in reverse sequential order. As shown, in step s31 the system runs the following set of smoothing filter equations on the current speech sample being processed together with the stored Kalman filter variables computed for the current speech sample being processed:

$$
\begin{aligned}
C(t) &= \sigma_e^2 (I - \sigma_e^2 U(t)) \\
\underline{\eta}(t) &\sim N(0, C(t)) \\
V(t) &= \sigma_e^2 U(t) L(t) \\
\underline{r}(t-1) &= \underline{h}\, d(t)^{-1} w(t) + L(t)^T \underline{r}(t) - V(t)^T C(t)^{-1} \underline{\eta}(t) \\
U(t-1) &= \underline{h}\, d(t)^{-1} \underline{h}^T + L(t)^T U(t) L(t) + V(t)^T C(t)^{-1} V(t) \\
\underline{\tilde{e}}(t) &= \sigma_e^2 \underline{r}(t) + \underline{\eta}(t) \qquad \text{where } \underline{\tilde{e}}(t) = [\tilde{e}(t)\ \tilde{e}(t-1)\ \tilde{e}(t-2)\ \ldots\ \tilde{e}(t-r+1)]^T \\
\underline{\hat{s}}(t) &= \tilde{A}\, \underline{\hat{s}}(t-1) + \underline{\hat{e}}(t) \qquad \text{where } \underline{\hat{s}}(t) = [\hat{s}(t)\ \hat{s}(t-1)\ \hat{s}(t-2)\ \ldots\ \hat{s}(t-r+1)]^T \\
&\qquad\qquad \text{and } \underline{\hat{e}}(t) = [\tilde{e}(t)\ \ 0\ \ 0\ \ \ldots\ \ 0\ ]^T
\end{aligned}
\tag{34}
$$

where $\underline{\eta}(t)$ is a sample drawn from a Gaussian distribution having zero mean and covariance matrix C(t); the initial vector $\underline{r}$(t=N) and the initial matrix U(t=N) are both set to zero; and $\underline{s}$(0) is obtained from the processing of the previous frame (or if there are no previous frames can be set equal to zero). The processing then proceeds to step s33 where the estimate of the process noise ($\tilde{e}$(t)) for

the current speech sample being processed and the estimate of the raw speech sample ($\hat{s}(t)$) for the current speech sample being processed are stored. The processing then proceeds to step s35 where the system determines whether or not all the speech samples in the current frame have been processed. If they have not, then the processing proceeds to step s37 where the time variable t is decremented by one so that the previous sample in the current frame will be processed in the same way. Once all N samples in the current frame have been processed in this way and the corresponding process noise and raw speech samples have been stored, the second stage of the Simulation Smoother is complete and an estimate of $\underline{s}(n)$ will have been generated.

As shown in equations (4) and (8), the matrix S and the matrix Y require raw speech samples s(n-N-1) to s(n-N-k+1) and s(n-N-1) to s(n-N-r+1) respectively in addition to those in $\underline{s}(n)$. These additional raw speech samples can be obtained either from the processing of the previous frame of speech or if there are no previous frames, they can be set to zero. With these estimates of raw speech samples, the Gibbs sampler can be run to draw samples from the above described probability density functions.

Statistical Analysis Unit - Operation

A description has been given above of the theory underlying the statistical analysis unit 21. A description will now be given with reference to Figures 6 to 8 of the operation of the statistical analysis unit 21 that is used in the embodiment.

Figure 6 is a block diagram illustrating the principal components of the statistical analysis unit 21 of this embodiment. As shown, it comprises the above described Gibbs sampler 41, Simulation Smoother 43 (including the Kalman filter 43-1 and smoothing filter 43-2) and model order selector 45. It also comprises a memory 47 which receives the speech samples of the current frame to be processed, a data analysis unit 49 which processes the data generated by the Gibbs sampler 41 and the model order selector 45 and a controller 50 which controls the operation of the statistical analysis unit 21.

As shown in Figure 6, the memory 47 includes a non volatile memory area 47-1 and a working memory area 47-2. The non volatile memory 47-1 is used to store the joint probability density function given in equation (19) above and the equations for the variances and mean values and the equations for the Inverse Gamma parameters given

above in equations (22) to (24) and (27) to (30) for the above mentioned conditional probability density functions for use by the Gibbs sampler 41. The non volatile memory 47-1 also stores the Kalman filter equations given above in equation (33) and the smoothing filter equations given above in equation 34 for use by the Simulation Smoother 43.

Figure 7 is a schematic diagram illustrating the parameter values that are stored in the working memory area (RAM) 47-2. As shown, the RAM includes a store 51 for storing the speech samples $y_f(1)$ to $y_f(N)$ output by the analogue to digital converter 17 for the current frame (f) being processed. As mentioned above, these speech samples are used in both the Gibbs sampler 41 and the Simulation Smoother 43. The RAM 47-2 also includes a store 53 for storing the initial estimates of the model parameters (g=0) and the M samples (g = 1 to M) of each parameter drawn from the above described conditional probability density functions by the Gibbs sampler 41 for the current frame being processed. As mentioned above, in this embodiment, M is 100 since the Gibbs sampler 41 performs 150 iterations on each frame of input speech with the first fifty samples being discarded. The RAM 47-2 also includes a store 55 for storing W(t), d(t) and

L(t) for t = 1 to N which are calculated during the processing of the speech samples in the current frame of speech by the above described Kalman filter 43-1. The RAM 47-2 also includes a store 57 for storing the estimates of the raw speech samples ($\hat{s}_f(t)$) and the estimates of the process noise ($\tilde{e}_f(t)$) generated by the smoothing filter 43-2, as discussed above. The RAM 47-2 also includes a store 59 for storing the model order counts which are generated by the model order selector 45 when the model orders for the AR filter model and the channel model are updated.

Figure 8 is a flow diagram illustrating the control program used by the controller 50, in this embodiment, to control the processing operations of the statistical analysis unit 21. As shown, in step s41, the controller 50 retrieves the next frame of speech samples to be processed from the buffer 19 and stores them in the memory store 51. The processing then proceeds to step s43 where initial estimates for the channel model, raw speech samples and the process noise and measurement noise statistics are set and stored in the store 53. These initial estimates are either set to be the values obtained during the processing of the previous frame of speech or, where there are no previous frames of speech,

are set to their expected values (which may be zero). The processing then proceeds to step s45 where the Simulation Smoother 43 is activated so as to provide an estimate of the raw speech samples in the manner described above. The processing then proceeds to step s47 where one iteration of the Gibbs sampler 41 is run in order to update the channel model, speech model and the process and measurement noise statistics using the raw speech samples obtained in step s45. These updated parameter values are then stored in the memory store 53.

The processing then proceeds to step s49 where the controller 50 determines whether or not to update the model orders of the AR filter model and the channel model. As mentioned above, in this embodiment, these model orders are updated every third Gibbs iteration. If the model orders are to be updated, then the processing proceeds to step s51 where the model order selector 45 is used to update the model orders of the AR filter model and the channel model in the manner described above. If at step s49 the controller 50 determines that the model orders are not to be updated, then the processing skips step s51 and the processing proceeds to step s53. At step s53, the controller 50 determines whether or not to perform another Gibbs iteration. If another iteration is

to be performed, then the processing proceeds to decision block s55 where the controller 50 decides whether or not to update the estimates of the raw speech samples (s(t)). If the raw speech samples are not to be updated, then the processing returns to step s47 where the next Gibbs iteration is run.

As mentioned above, in this embodiment, the Simulation Smoother 43 is run every fourth Gibbs iteration in order to update the raw speech samples. Therefore, if the controller 50 determines, in step s55 that there has been four Gibbs iterations since the last time the speech samples were updated, then the processing returns to step s45 where the Simulation Smoother is run again to provide new estimates of the raw speech samples (s(t)). Once the controller 50 has determined that the required 150 Gibbs iterations have been performed, the controller 50 causes the processing to proceed to step s57 where the data analysis unit 49 analyses the model order counts generated by the model order selector 45 to determine the model orders for the AR filter model and the channel model which best represents the current frame of speech being processed. The processing then proceeds to step s59 where the data analysis unit 49 analyses the samples drawn from the conditional densities by the Gibbs sampler

41 to determine the AR filter coefficients (a), the channel model coefficients (h), the variances of these coefficients and the process and measurement noise variances which best represent the current frame of speech being processed. The processing then proceeds to step s61 where the controller 50 determines whether or not there is any further speech to be processed. If there is more speech to be processed, then processing returns to step S41 and the above process is repeated for the next frame of speech. Once all the speech has been processed in this way, the processing ends.

Data Analysis unit

A more detailed description of the data analysis unit 49 will now be given with reference to Figure 9. As mentioned above, the data analysis unit 49 initially determines, in step s57, the model orders for both the AR filter model and the channel model which best represents the current frame of speech being processed. It does this using the counts that have been generated by the model order selector 45 when it was run in step s51. These counts are stored in the store 59 of the RAM 47-2. In this embodiment, in determining the best model orders, the data analysis unit 49 identifies the model order having the highest count. Figure 9a is an exemplary

histogram which illustrates the distribution of counts that is generated for the model order (k) of the AR filter model. Therefore, in this example, the data analysis unit 49 would set the best model order of the AR filter model as five. The data analysis unit 49 performs a similar analysis of the counts generated for the model order (r) of the channel model to determine the best model order for the channel model.

Once the data analysis unit 49 has determined the best model orders (k and r), it then analyses the samples generated by the Gibbs sampler 41 which are stored in the store 53 of the RAM 47-2, in order to determine parameter values that are most representative of those samples. It does this by determining a histogram for each of the parameters from which it determines the most representative parameter value. To generate the histogram, the data analysis unit 49 determines the maximum and minimum sample value which was drawn by the Gibbs sampler and then divides the range of parameter values between this minimum and maximum value into a predetermined number of sub-ranges or bins. The data analysis unit 49 then assigns each of the sample values into the appropriate bins and counts how many samples are allocated to each bin. It then uses these counts to

calculate a weighted average of the samples (with the weighting used for each sample depending on the count for the corresponding bin), to determine the most representative parameter value (known as the minimum mean square estimate (MMSE)). Figure 9b illustrates an example histogram which is generated for the variance (σ_e^2) of the process noise, from which the data analysis unit 49 determines that the variance representative of the sample is 0.3149.

In determining the AR filter coefficients (a_i for i = i to k), the data analysis unit 49 determines and analyses a histogram of the samples for each coefficient independently. Figure 9c shows an exemplary histogram obtained for the third AR filter coefficient (a_3), from which the data analysis unit 49 determines that the coefficient representative of the samples is -0.4977.

In this embodiment, the data analysis unit 49 only outputs the AR filter coefficients which are passed to the channel encoder 71 shown in Figure 1. The AR filter coefficients (and the remaining parameter values determined by the data analysis unit 49) are also stored in the RAM 47-2 for use during the processing of the next frame of speech.

As the skilled reader will appreciate, a speech processing technique has been described above which uses statistical analysis techniques to determine sets of AR filter coefficients representative of an input speech signal. The technique is more robust and accurate than prior art techniques which employ maximum likelihood estimators to determine the AR filter coefficients. This is because the statistical analysis of each frame uses knowledge obtained from the processing of the previous frame.

In addition, with the analysis performed above, the model order for the AR filter model is not assumed to be constant and can vary from frame to frame. In this way, the optimum number of AR filter coefficients can be used to represent the speech within each frame. As a result, the AR filter coefficients output by the statistical analysis unit 21 will more accurately represent the corresponding input speech. In contrast, with the prior art linear prediction systems, the number of AR coefficients is assumed to be constant and hence these prior art techniques tend to over parametrise the speech in order to ensure that information is not lost. As a result, with the statistical analysis described above, the amount of data which has to be transmitted from the

transmitter to the receiver will be less than with the prior art systems which assume a fixed size of AR filter model.

Further still, since the underlying process model that is used separates the speech source from the channel, the AR filter coefficients that are determined will be more representative of the actual speech and will be less likely to include distortive effects of the channel.

Further still, since variance information is available for each of the parameters, this provides an indication of the confidence of each of the parameter estimates. This is in contrast to maximum likelihood and least squares approaches, such as linear prediction analysis, where point estimates of the parameter values are determined.

Alternative Embodiments

In the above embodiment, the statistical analysis unit was used in order to parameterise the input speech signal for onward transmission to a remote receiver. It also generated a number of other parameter values (such as the process noise variances and the channel model coefficients), but these were not output by the

09665585.053001

statistical analysis unit. As those skilled in the art will appreciate, the AR coefficients and some of the other parameters which are calculated by the statistical analysis unit can also be used in the transmission system. For example, the variance information of the AR coefficients can be used to control the type of channel encoding which is employed by the channel encoder 71, since this variance information is indicative of the quality of the input speech signal. For example, different channel encoding techniques may be used in dependence upon the quality of the input speech signal, with the particular technique being used depending upon the quality of the speech within each frame.

In the above embodiment, the speech generation process was modelled as an auto-regressive (AR) process and the channel was modelled as a moving average (MA) process. As those skilled in the art will appreciate, other signal models may be used. However, these models are preferred because it has been found that they suitably represent the speech source and the channel they are intended to model.

In the above embodiment, a speech quality indicator was determined for each frame of speech that is processed and

09865585.053001

the channel encoder 71 encodes each frame of speech in dependence upon the quality measure determined for each frame. As those skilled in the art will appreciate, this is not essential. For example, a moving average quality indicator may be determined and the channel encoder may be arranged to change its encoding technique when the moving average goes above or below a predetermined threshold value.

In the above embodiments, Gaussian and Inverse Gamma distributions were used to model the various prior probability density functions of equation (19). As those skilled in the art of statistical analysis will appreciate, the reason these distributions were chosen is that they are conjugate to one another. This means that each of the conditional probability density functions which are used in the Gibbs sampler will also either be Gaussian or Inverse Gamma. This therefore simplifies the task of drawing samples from the conditional probability densities. However, this is not essential. The noise probability density functions could be modelled by Laplacian or student-t distributions rather than Gaussian distributions. Similarly, the probability density functions for the variances may be modelled by a distribution other than the Inverse Gamma distribution.

09865585.053001

For example, they can be modelled by a Rayleigh distribution or some other distribution which is always positive. However, the use of probability density functions that are not conjugate will result in increased complexity in drawing samples from the conditional densities by the Gibbs sampler.

Additionally, whilst the Gibbs sampler was used to draw samples from the probability density function given in equation (19), other sampling algorithms could be used. For example the Metropolis-Hastings algorithm (which is reviewed together with other techniques in a paper entitled "Probabilistic inference using Markov chain Monte Carlo methods" by R. Neal, Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto, 1993) may be used to sample this probability density.

In the above embodiment, a Simulation Smoother was used to generate estimates for the raw speech samples. This Simulation Smoother included a Kalman filter stage and a smoothing filter stage in order to generate the estimates of the raw speech samples. In an alternative embodiment, the smoothing filter stage may be omitted, since the Kalman filter stage generates estimates of the raw speech

(see equation (33)). However, these raw speech samples were ignored, since the speech samples generated by the smoothing filter are considered to be more accurate and robust. This is because the Kalman filter essentially generates a point estimate of the speech samples from the joint probability density function $p(\underline{s}(n)|\underline{a},k,\sigma_e^2)$, whereas the Simulation Smoother draws a sample from this probability density function.

In the above embodiment, a Simulation Smoother was used in order to generate estimates of the raw speech samples. It is possible to avoid having to estimate the raw speech samples by treating them as "nuisance parameters" and integrating them out of equation (19). However, this is not preferred, since the resulting integral will have a much more complex form than the Gaussian and Inverse Gamma mixture defined in equation (19). This in turn will result in more complex conditional probabilities corresponding to equations (20) to (30). In a similar way, the other nuisance parameters (such as the coefficient variances or any of the Inverse Gamma, alpha and beta parameters) may be integrated out as well. However, again this is not preferred, since it increases the complexity of the density function to be sampled using the Gibbs sampler. The technique of integrating

09865585.053001

out nuisance parameters is well known in the field of statistical analysis and will not be described further here.

In the above embodiment, the data analysis unit analysed the samples drawn by the Gibbs sampler by determining a histogram for each of the model parameters and then determining the value of the model parameter using a weighted average of the samples drawn by the Gibbs sampler with the weighting being dependent upon the number of samples in the corresponding bin. In an alterative embodiment, the value of the model parameter may be determined from the histogram as being the value of the model parameter having the highest count. Alternatively, a predetermined curve (such as a bell curve) could be fitted to the histogram in order to identify the maximum which best fits the histogram.

In the above embodiment, the statistical analysis unit modelled the underlying speech production process with a separate speech source model (AR filter) and a channel model. Whilst this is the preferred model structure, the underlying speech production process may be modelled without the channel model. In this case, there is no need to estimate the values of the raw speech samples

using a Kalman filter or the like, although this can still be done. However, such a model of the underlying speech production process is not preferred, since the speech model will inevitably represent aspects of the channel as well as the speech. Further, although the statistical analysis unit described above ran a model order selection routine in order to allow the model orders of the AR filter model and the channel model to vary, this is not essential. In particular, the model order of the AR filter model and the channel model may be fixed in advance, although this is not preferred since it will inevitably introduce errors into the representation.

In the above embodiments, the speech that was processed was received from a user via a microphone. As those skilled in the art will appreciate, the speech may be received from a telephone line or may have been stored on a recording medium. In this case, the channel model will compensate for this so that the AR filter coefficients representative of the actual speech that has been spoken should not be significantly affected.

In the above embodiments, during the running of the model order selection routine, a new model order was proposed by drawing a random variable from a predetermined

Laplacian distribution function. As those skilled in the art will appreciate, other techniques may be used. For example the new model order may be proposed in a deterministic way (ie under predetermined rules), provided that the model order space is sufficiently sampled.

09856585.052001